PART2　シミュレーション（プログラミング）

目標
- 確定的モデルや確率的モデルのシミュレーションのプログラムが書けるようになる。

Chapter1　シミュレーション（プログラミング）

確定的モデルのシミュレーション

- **売上予測（数値表示）プログラム**

例えば，1個750円のメロンを販売する場合を考えてみる。

1個売れれば750円，2個売れれば1500円の売上となる。

商品単価が同じ場合，売上金額は以下の計算式で表すことができる。

売上金額　＝　商品単価　×　売れた個数

Python では，以下のように表すことができる。

```
tanka = 750
for i in range(1, 11): # 開始値 :1, 終了値 :11
  uriage = tanka * i
  print(str(i)  + '個の売上：'+ str(uriage) + '円')
```

復習

確定的モデルとは，変動する要素がなく，結果が1つに定まるモデルのこと。

覚えておこう

range(1,11) は数式で表現すると 1 ≦ i <11 の意味で 11 は含まないことに注意する。

実行結果

1個の売上：750円
2個の売上：1500円
3個の売上：2250円
4個の売上：3000円
5個の売上：3750円
6個の売上：4500円
7個の売上：5250円
8個の売上：6000円
9個の売上：6750円
10個の売上：7500円

• 売上予測（グラフ表示）プログラム

売上予測を，視覚的に理解しやすくするため，折れ線グラフで表現する。グラフ描画には〔①　　　　　　　　〕Uriage にグラフ描画するための値を格納し，〔②　　　　　　　　　　　〕モジュールを利用する。

知っておこう

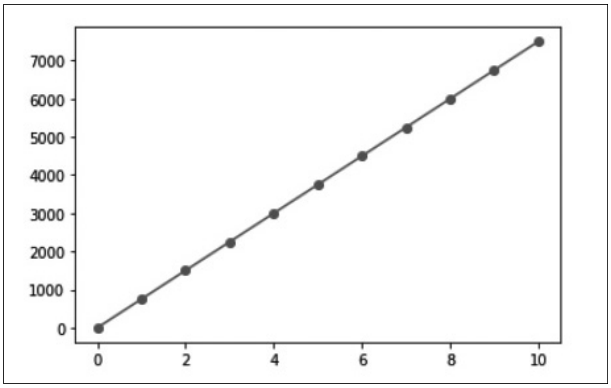
matplotlib.pyplot はグラフを描くのに特化したモジュールである。

```
import matplotlib.pyplot as plt

tanka = 750
Uriage = [0,0,0,0,0,0,0,0,0,0,0]
for i in range(len(Uriage)):
  Uriage[i] = tanka * i
plt.plot(Uriage, marker='o') #売上グラフをプロット
```

実行結果

```
[<matplotlib.lines.Line2D at 0x7fd425148c50>]
```



確率的モデルのシミュレーション

• 乱数（擬似乱数）とモンテカルロ法

乱数とは，ある一定の範囲内において，全ての数が同じ確率で現れるような数のことである。ただし，コンピュータが作る乱数は，計算で値を生成しているため，実際には完全にランダムではなく，擬似的な乱数となっている。これを〔③　　　　〕（以後「乱数」）という。確率的モデルの中で乱数を用いて問題を解決する手法を〔④　　　　　　　　　〕という。

復習

確率的モデルとは，変動する要素があり，結果が１つに定まらないモデルのこと。

- **乱数の生成方法**

Python で乱数を扱う場合は，〔⑤　　　　　　　　〕モジュールを用いる。

例えば，1 以上 6 以下の整数の乱数を生成して表示する場合は，次のように表現できる。

```
import random

r = random.randint(1, 6) # 1 以上 6 以下の整数の乱数を生成
print(r)
```

このプログラムは，実行するたびに 1 から 6 までのいずれかの数がランダムに表示される。

- **サイコロの出目を調べるプログラム**

乱数を用いてサイコロのプログラムを作成する。1 から 6 までの整数の乱数を生成させる操作を 60 回行い，分布を表示する。

```
import random

shikoukaisuu = 60
Kaisuu = [0,0,0,0,0,0]
for num in range(shikoukaisuu):
  r = random.randint(1, 6) # 1 以上 6 以下の整数の乱数を生成
  Kaisuu[r - 1] = Kaisuu[r - 1] + 1
print(Kaisuu)
```

実行結果

```
[17, 8, 6, 8, 12, 9]
```

試行回数 shikoukaisuu を 600，6000，60000 と大きくすればするほど，それぞれの目が出る回数が均等になっていくことが分かる。

まとめ

・Pythonの**matplotlib.pyplot**というグラフ描画モジュールを利用することで，シミュレーション結果などをグラフで視覚的に表現することができる。
・ある一定の範囲内において，全ての数が同じ確率で現れるような数のことを**乱数**という。Python では，**random.randint**関数を用いることで乱数を生成することができる。

Chapter2　実践問題

問１　お風呂の水量（L）を１分当たりの注水量（L）と時間（分）から求めるため，次のプログラムを作成した。

```
tyuusuiryo = 9 # 1分当たりの注水量（L）
for i in range(1, 6, 2): # 開始値 :1，終了値 :6，間隔値 :2
  suiryo = （ア）
  print(str(i) + '分後の水量：' + str(suiryo) +'L')
```

(1)　水量と注水量と時間の関係式を答えなさい。

(2)　空欄（ア）にあてはまる数式を答えなさい。

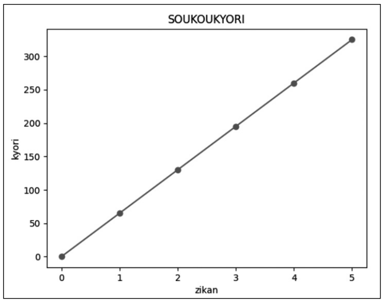
(3)　このプログラムの実行結果を答えなさい。

問２　自動車の走行距離（km）を時速と時間から求めて，その関係をグラフで表現するために，次のプログラムを作成した。コメントなどをヒントに，出力結果のグラフと同等の結果になるように，空欄（ア），（イ）を埋めて，プログラムを完成させなさい。

```
import matplotlib.pyplot as plt

zisoku = 65 # 1時間当たりの走行距離（km）
Soukoukyori = [0,0,0,0,0,0]
（ア） # 開始値 :0，終了値 :6，間隔値 :1
  （イ） # 走行距離の計算と代入

plt.title('SOUKOUKYORI') # グラフのタイトル
plt.xlabel('zikan') # 横軸のラベル
plt.ylabel('kyori') # 縦軸のラベル
plt.plot(Soukoukyori, marker='o') # グラフをプロット
```



問3　乱数を用いたおみくじのプログラムを作成した。コメントなどをヒントにして，空欄（ア），（イ）を埋めて，プログラムを完成させなさい。なお，おみくじの結果は，必ず表示されるものとする。

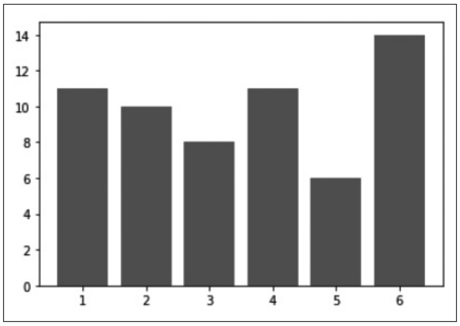
```
import random

[    (ア)    ] # 0 以上 5 以下の整数の乱数を生成
if r == 0:
  print(' 大吉 ')
elif r == 1:
  print(' 中吉 ')
elif r == 2:
  print(' 小吉 ')
[    (イ)    ]
  print(' 吉 ')
```

問4　乱数を用いて，サイコロのプログラムを作成した。このプログラムでは，1 から 6 までの整数の乱数を生成して，それを 60 回試行して，分布を表示している。コメントなどをヒントにして，空欄（ア），（イ）を埋めて，プログラムを作成しなさい。

```
import [    (ア)    ] as plt
import random

shikoukaisuu = 60
Saikoro = [1,2,3,4,5,6] # グラフの横軸（x 軸）
Kaisuu = [0,0,0,0,0,0]
for num in range([ (イ) ]):
  r = random.randint(1, 6) # 1 以上 6 以下の整数の乱数を生成
  Kaisuu[r - 1] = Kaisuu[r - 1] + 1
plt.bar(Saikoro, Kaisuu)
```



第 7 講　PART2　確認問題

問 1　モンテカルロ法の説明として最も適切なものを，次の①～④のうちから選びなさい。

① 同じデータが連続する部分に注目して圧縮する手法
② シミュレーションに乱数を用いて問題解決する手法
③ 計画，実行，評価，改善を繰り返す手法
④ 並列に行われる処理を図形や線などを用いて，視覚的に表す手法

問 2　家に 10 L（リットル）のジュースがあるとする。日によって飲むジュースの量を，1L・2L・3L のいずれかにする場合に，ジュースを飲み切るのにかかる日数をシミュレーションする Python のプログラムについて，以下の問いに答えなさい。

```
import random

juice = 10 #家にある量（L）
nissu = 0
while juice > 0:
  r = random.randint(1, 3)
  [      (ア)      ]
  nissu = nissu + 1

print(' ジュースを飲み切るには '+ str(nissu) + ' 日かかる ')
```

(1)　空欄（ア）に入るものとして最も適切なものを，次の①～④のうちから一つ選びなさい。

① juice = r　　② juice = juice + r
③ juice = juice - r　　④ juice = juice * r

(2) ジュースを全く飲まない日がある場合も想定してシミュレーションできるようにする際，プログラムをどのように変更するべきかを，次の①～④のうちから選びなさい。

① 変数 r に代入する値を r ＝ random.randint(0, 3) にする

② 変数 r に代入する値を r ＝ random.randint(1, 4) にする

③ while juice ＞ 0: を while juice => 0: にする

④ while juice ＞ 0: を while juice ＞ 1: にする

第７講　モデル化とシミュレーション

PART1　モデル化とシミュレーション

問１　（ア）③　（イ）①　（ウ）②　（エ）①

問２　（ア）③　（イ）③　（ウ）①　（エ）②

問３　（ア）②　（イ）①　（ウ）④　（エ）③

問４　②→③→①

問５　③

PART2　シミュレーション（プログラミング）

問１　(1)　水量＝注水量×時間　(2)　tyuusuiryo * i

(3)　１分後の水量：9L

３分後の水量：27L

５分後の水量：45L

問２　（ア）for i in range(6):　（イ）Soukoukyori[i] = zisoku * i

問３　（ア）r = random.randint(0, 5)　（イ）else:

問４　（ア）matplotlib.pyplot　（イ）shikoukaisuu

PART3　シミュレーション（表計算）

問１　(1)　省略　(2)　省略　(3)　省略

問２　省略

第 7 講　PART2　確認問題　解答

問 1　②

解説：
確率的モデルのシミュレーションにおいて，乱数を用いて問題を解決する手法をモンテカルロ法という。
①はランレングス法，③は PDCA モデル，④はアクティビティ図の説明である。

問 2

(1)　③

解説：
毎日飲むジュースを乱数によって決めているため，家にあるジュースの量を表す変数 juice が 0 になるまで，生成した乱数(変数 r)を減算していく必要がある。したがって③が正しい。

(2)　①

解説：
ジュースを全く飲まない日も含めるには，生成する乱数が 1，2，3 のどれかだったのを，0，1，2，3 に変更すればよい。0 も含めて乱数を生成するには，randam.randint の引数を「0,3」にする必要がある。したがって①が正しい。